

EUR 2760.f

COMMUNAUTE EUROPEENNE DE L'ENERGIE ATOMIQUE - EURATOM

FORTRAN INPUT - OUTPUT PACKAGE IN TRAP-MODE

par

J. PIRE et L. SANGERMANO

1966



Centre Commun de Recherche Nucléaire
Etablissement d'Ispira - Italie

Centre de Traitement de l'Information Scientifique - CETIS

AVERTISSEMENT

Le présent document a été élaboré sous les auspices de la Commission de la Communauté Européenne de l'Energie Atomique (EURATOM).

Il est précisé que la Commission d'EURATOM, ses contractants, ou toute personne agissant en leur nom :

ne garantissent pas l'exactitude ou le caractère complet des informations contenues dans ce document, ni que l'utilisation d'une information, d'un équipement, d'une méthode ou d'un procédé quelconque décrits dans le présent document ne porte pas atteinte à des droits privés ;

n'assument aucune responsabilité pour les dommages qui pourraient résulter de l'utilisation d'informations, d'équipements, de méthodes ou procédés divulgués dans le présent document.

Ce rapport est vendu dans les bureaux de vente indiqués en 4^e page de couverture

au prix de FF 4,—	FB 40,—	DM 3.20	Lit. 500	Fl. 3,—
-------------------	---------	---------	----------	---------

Prière de mentionner, lors de toute commande, le numéro EUR et le titre qui figurent sur la couverture de chaque rapport.

Imprimé par L. Vanmelle, Gent
Bruxelles, mars 1966

Le présent document a été reproduit à partir de la meilleure copie disponible.

EUR 2760.f

COMMUNAUTE EUROPEENNE DE L'ENERGIE ATOMIQUE - EURATOM

FORTRAN INPUT - OUTPUT PACKAGE IN TRAP-MODE

par

J. PIRE et L. SANGERMANO

1966



Centre Commun de Recherche Nucléaire
Etablissement d'Ispra - Italie

Centre de Traitement de l'Information Scientifique - CETIS

TABLE DES MATIERES

Introduction	Pg.	3
Description Générale		3
Incompatibilités avec FORTRAN		5
Usage du système par les programmes écrits en FORTRAN		6
Routines spéciales BCDRD, BCDWR, BLOCKED OUTPUT		9
IOT Entrées normales		11
IOT Entrée spéciales		
Fonctionnement du Système		16
Liste des LABEL - ENTREES - LONGUEUR - EXTERNAL		22
Routines à insérer		24

RESUME

F.I.O.T. est un système de sous-programmes de lecture-écriture pour l'ordinateur IBM 7090 travaillant en trap-mode et compatible avec la version 3 de Fortran II.

Les programmeurs désirant utiliser F.I.O.T. n'ont que très peu de modifications à apporter à leur programmes écrits en Fortran.

Les gains de temps constatés en appliquant ce système à divers programmes non conçus spécialement est de l'ordre de 15 %.

FORTRAN INPUT - OUTPUT PACKAGE

IN TRAP-MODE

INTRODUCTION (°)

L'ensemble des routines constituant FIOT (FORTRAN INPUT OUTPUT PACKAGE IN TRAP-MODE) a été écrit afin de faire bénéficier les programmes écrits en FORTRAN II et s'exécutant sous le contrôle de la version 3 (en IBSYS) des possibilités qu'offre le "trapping" de canal effectuant une partie des calculs en superposition de temps avec les entrées de données et écritures des résultats. De plus, optionnellement, l'écriture sur la bande normale de sortie peut être exécutée en "blocking" réunissant plusieurs lignes en un seul enregistrement.

En ce qui concerne la superposition du calcul et de l'écriture les routines normales sont assez efficaces, sauf si de longs enregistrements doivent être écrits (mode binaire); en ce qui concerne la lecture, au contraire, elles ne permettent aucune superposition. FIOT apporte à FORTRAN un système de manipulation des unités périphériques analogue à un IOCS réduit à l'ombre de lui-même et n'exigeant pour être utilisé que très peu de modifications du programme écrit pour le système usuel.

Généralement, seul le programme principal et, en cas de chain, les programmes principaux doivent être recompilés. A titre d'exemple pour modifier SIFT il est nécessaire de recompiler 6 routines sur la soixantaine que comporte le programme.

DESCRIPTION GENERALE

Essentiellement, FIOT se compose de deux ensembles bien distincts:

- 1° un système de contrôle d'unités périphériques et de buffers utilisant IOEX (IOT)

(°) Manuscrit reçu le 26 janvier 1966

2° un ensemble de routines comportant les entrées homonymes de celles des routines utilisées par FORTRAN pour l'exécution de ses statements de lecture et d'écriture; cet ensemble assure la compatibilité entre FORTRAN et la routine IOT.

De toutes les routines originales du FORTRAN, seule (IOH) a été conservée intacte; certaines sont devenues des sous-routines muettes (IOS se résume à TRA 1, 4), d'autres sont complètement disparu.

Disons tout de suite que le paquet complet (y compris IOT) occupe moins de mémoires (environ 700_g de moins) que le paquet FORTRAN normal. Il ne comporte cependant aucun buffer propre (alors que (IOB) normal en comporte 2).

Tous les buffers doivent être réservés par le programmeur et mis à disposition du système à l'aide du statement

CALL OPEN (.....)

Ce statement avec la réservation des buffers, sont par ailleurs les seules modifications qui doivent nécessairement être introduites dans les programmes FORTRAN pour les adapter à FIOT.

L'appel CALL CLOSE (.....) peut éventuellement permettre de désactiver momentanément une unité et d'assigner les buffers qui lui appartiennent à une autre unité. Les conventions adoptées permettent de réactiver l'unité au point où on l'avait désactivée.

Liste des subroutines et points d'entrée.

A) PAQUET FORTRAN

(BST)

CHAIN

(EXEM)

EXIT, EXESEL, EXTTFN, FLGXIT

(IOB), (SET), (BUF), (EXB)
(IOS)
(IOU)
PAUSE
(RER)
(SPH)
(STB), (WLR)
(STH), (STHM), (TES) (STHD)
(TSB), (RLR), (IND)
(TSH), (TSHM), (TSHD), (FHW)*, SETOUT*
(DUM)

L'appel aux sous-routines ci-dessus est strictement le même qu'en FORTRAN.

INCOMPATIBILITES AVEC FORTRAN

- 1) La principale incompatibilité avec FORTRAN est l'impossibilité de faire le BACKSPACE de plus de un record logique sur une unité binaire.

Ce statement est si rarement utilisé et la procédure classique tellement longue que nous n'avons pas hésité à y substituer une méthode directe (conservation du nombre de records physique du dernier record logique) mais applicable à un seul backspace.

Nous nous excusons auprès de ceux qui font pratique courante des BACKSPACE en chaîne. ⁽¹⁾

- 2) Le statement "PAUSE" compilé par FORTRAN ne réussit pas à arrêter le déroulement du programme travaillant en "trapping" de canal.
Il est nécessaire d'y substituer un CALL PAUSE (n)

- 3) Le statement PRINT ne peut utiliser un FORMAT de plus de 80

* Dans la version effectuant le blocking.

- (1) Une seconde version est cependant disponible éliminant cette incompatibilité.

caractères par ligne. L'usage de ce statement devant normalement être limité à l'envoi de message à l'opérateur, nous ne croyons pas que cette restriction soit très importante.

- 4) Le statement PUNCH écrit automatiquement les images des cartes en BCD sur SYSP1. Un end-of-file est automatiquement écrit sur cette unité par EXIT si un statement PUNCH a été exécuté.
- 5) L'exécution doit se terminer soit par EXIT soit par EXEM
- 6) PDUMP n'existe pas; le système comporte uniquement le dump d'IBSYS.

USAGE DU SYSTEME PAR DES PROGRAMMES ECRITS EN FORTRAN

Afin d'assurer la simultanéité des lectures, écritures et calcul le système doit disposer de "buffers" pour y lire à l'avance les bandes définies "en lecture" et d'autres buffers pour y sauver momentanément l'information devant être écrit sur bande.

Le système FORTRAN II ne prévoyant rien de semblable, et ne construisant aucune information générale au sujet des unités périphériques utilisées, il appartient au programmeur de renseigner FIOT à ce sujet. La subroutine OPEN fourni au système toutes les informations nécessaires au sujet d'une unité périphérique.

```
CALL OPEN (IUNIT, IRW, BUF, LBUF, LREC, FLAG)
```

où

UNIT contient le numéro logique de l'unité considérée de 1 à 16;
voir (IOU)

IRW est un code décrivant la fonction de l'unité et se composant comme suit

+ = lecture
- = écriture
0 = mode binaire
7 = mode BCD

Les combinaisons possibles sont donc:

7 lecture BCD
-7 écriture BCD
0 lecture binaire
-0 écriture mode binaire

BUF est le nom d'un vecteur mis par le programmeur à l'entière disposition du système; le programmeur ne peut pas s'en servir tant que le système l'utilise. Nous verrons par la suite quand le système remettra BUF à disposition du programmeur. BUF doit évidemment avoir été réservé à l'aide d'un statement DIMENSION d'une longueur au moins égale à LBUF.

LBUF est la dimension de BUF

LREC est la longueur maximum des enregistrements physiques à traiter sur cette unité par exemple si l'unité contient des images de carte B.C.D., LREC vaut 14; si l'unité est l'unité normale d'écriture LREC = 22; si l'unité est en mode binaire LREC = 256 généralement. Si on choisit la version avec blocking de l'écriture sur l'unité normale de sortie, LREC peut être de l'ordre de 250 mots (plus grand ou égal à 22; plus petit que 257 à Ispra - Le maximum est imposé par la capacité de la 1401 chargée de l'impression). Cependant en binaire, si la longueur L du record logique le plus long est inférieure à 256 mots rien n'empêche de faire LREC = L+1 (+1: pour le mot que FORTRAN ajoute au début des enregistrements binaires écrits par le statement WRITE TAPE.

FLAG est une mémoire mise par le programmeur à disposition de FIOT. Le système la remettra à disposition du programmeur en même temps que BUF. FIOT y place des informations au sujet du record actuellement traité par le programme prin-

cipal. Nous reviendrons sur le contenu de cette mémoire.

Remarques importantes

LBUF doit être plus grand ou égal à $2*(LREC+1)$ et pour avoir une utilisation efficient de la mémoire, doit être un multiple de $(LREC+1)$.

Note IRW stipule le mode dans lequel la bande sera écrite ou lue et ce indépendamment du statement FORTRAN qui manipule l'unité. Il est donc possible d'écrire en mode binaire un enregistrement provenant d'un statement WRITE OUTPUT TAPE ou de lire par un statement READ INPUT TAPE un enregistrement écrit en mode binaire. Il est clair que ceci est une "possibilité" qu'offre le système et qu'elle ne doit être utilisée qu'avec précaution dans des cas très particulier. (Par exemple lecture de cartes binaires ou écriture de cartes pseudo-binaires (cartes BCD condensées pouvant contenir : 2 caractères de code dont le second doit être un 5 et 130 autres caractères libres soit en tout 22 mots; cette longueur maximum est imposée par IOH).

Toute unité utilisée dans un des statements READ TAPE, READ INPUT TAPE, WRITE TAPE, WRITE OUTPUT TAPE doit au paravant avoir été déclarée (ouverte) par un appel à OPEN et n'être pas "fermée" au moment de l'exécution du statement.

Une unité est automatiquement fermée lors de l'exécution d'un des statements

ENDFILE ou REWIND adressant cette unité; ou lors de l'exécution de

CALL CLOSE (IUNIT)

ou

CALL CHAIN (R,T)

La fermeture remet à disposition du programmeur BUF et FLAG; il peut à sa guise en disposer; cependant avant d'utiliser à nouveau

l'unité pour une opération de lecture ou d'écriture, il devra avoir réouvert l'unité.

La subroutine CLOSE positionne l'unité de bande à l'endroit où elle se serait trouvée au même moment dans un programme FORTRAN ne faisant pas usage de FIOT c'est à dire à la fin du dernier enregistrement "écrit" ou "lu" par le programme (et non par le système).

Pour le cas d'unité en lecture, le système effectue donc le nombre de backspace nécessaire pour positionner la bande correctement et en écriture il ne rend le contrôle que lorsque tout a été écrit.

Notes pratiques

- a) on ne peut "ouvrir" une unité alors qu'elle est déjà ouverte. (Erreur fatale - diagnostique et fin d'exécution).
- b) on peut "fermer" une file déjà fermée (c'est une petite perte de temps sans gravité)
- c) pour les unités de lecture que l'on désire fermer avant le dernier enregistrement il est préférable de ne pas donner un trop grand buffer (de ne pas permettre la lecture en avance de trop d'enregistrements) car il résulte une perte de temps en retour arrière sur cette unité.

L'expérience prouve qu'un buffer pouvant contenir de 2 à 4 enregistrements en lecture BCD est amplement suffisant.

ROUTINES SPECIALES

Deux routines spéciales utilisables directement en FORTRAN ont été adjointes à la librairie.

BCDRD et BCDWR permettent de lire et d'écrire des enregistrements en mode BCD aussi longs qu'on le désire. BCDWR permet

notamment le "BLOCKED OUTPUT" sans faire appel à la (STH) spéciale.

SETOUT permet de modifier le numéro de l'unité logique d'output normal sur laquelle se fait le blocking. Le standard du système est le numéro 6.

```
CALL SETOUT(N)
```

Description et usage

BCDRD

```
CALLBCDRD (IUNIT,LFORM)
```

```
READ n, List
```

IUNIT est le numéro logique de l'unité de lecture

LFORM est la longueur de l'enregistrement partiel décrit dans un format FORTRAN.

Un statement READ doit suivre immédiatement l'appel à BCDRD; il spécifie le format dont il est question ci-dessus et la liste des variables comme dans un appel ordinaire.

Pour chaque "enregistrement partiel" réclamé par le format, BCDRD fournira soit LFORM mots soit le résidu de l'enregistrement.

Après épuisement de l'enregistrement en cours il continuera avec le suivant.

Le résidu d'un enregistrement en cours est perdu lorsque la liste s'achève.

Le statement READ est un simple artifice forçant le compilateur à indiquer un FORMAT et à créer une liste de lecture facile à utiliser par le programmeur.

BCDWR

```
CALL BCDWR (IUNIT)
```

```
PRINT N, list
```

```
CALL FINWR
```

IUNIT est le numéro logique de l'unité sur laquelle il faut écrire.

Le statement PRINT sert à localiser le format et à géné-

rer la liste des variables à écrire.

CALL FINWR signale à BCDWR que l'enregistrement est terminé.

BCDWR stocke dans le buffer d'écriture les enregistrements partiels tels qu'ils sont décrits dans le format. Si une "ligne d'écriture" (enregistrement partiel) ne peut tenir dans l'espace encore disponible le buffer est écrit sur bande (à concurrence du nombre de mots utiles qu'il contient) et la nouvelle ligne est écrite dans un nouveau buffer. L'appel à FINWR provoque l'écriture du dernier buffer entamé.

Notons que le programmeur doit fournir lui-même le caractère qui au moment de l'impression en 1401 détermine la "fin de ligne" (72_8)

L'usage de BCDWR est un peu délicat et il vaut mieux lui préférer généralement l'usage de la (STH) de blocking.

IOT

Nous avons indiqué ci-dessus le type de modification à apporter à un programme écrit en FORTRAN pour l'adapter à l'usage de FIOT.

Voyons maintenant les possibilités qu'offre IOT pour un programmeur désirant utiliser l'entière capacité du système.

Entrées

Les entrées utilisables en FORTRAN sont:

CALL OPEN (IUNIT, IRW, BUF, LBUF, LREC, FLAG), voir explication plus haut

CALL CLOSE (IUNIT), voir explication plus haut

CALL CLOSAL, ferme toute les files ouvertes

CALL GET (IUNIT), met dans FLAG

- 1) en adresse l'adresse du buffer à lire ou à écrire par le programme principal;
- 2) en décrémente le nombre de mots utiles se trouvant dans le buffer en lecture; la longueur du buffer en écriture

3) dans le préfixe les "flags" suivants

- 2 si EOF
 - 1 si erreur permanente
 - 2 si End of Tape en écriture
- } en lecture

(Note : un statement FORTRAN : READ TAPE ou READ INPUT TAPE sur la même unité donné après un CALL GET (INPUT) s'exécutera sur le même buffer que celui pour lequel on a fourni le "flag" par GET.

Un CALL GET venant directement après un autre CALL GET fait avancer d'un pas l'utilisation des buffers c'est à dire que a) en écriture le contenu du buffer indiqué par le 1er GET sera écrit sur la bande, la longueur de l'enregistrement sera de LREC mots; b) en lecture, le contenu du buffer indiqué par le 1er GET est irrémédiablement perdu pour le programme (le buffer a été remis à disposition de l'unité de lecture qui probablement y introduit déjà de nouvelles données).

CALL SETPR (IUNIT)

Donne la précéance à une unité pour l'activation. La routine de précéance inclue dans le système cherche à mettre à disposition du programmeur au moins la moitié des "records" qu'il a défini, puis d'égaliser entre toutes les files le nombre de records non disponibles. Les files pour lesquelles on a défini des buffers contenant un petit nombre de records sont donc nettement désavantagées (c'est souvent le cas pour les unités définies en mode binaire). Les routine STB et BST utilisent la SETPR de façon à s'assurer le plus vite possible un "record" disponible; cependant si le programmeur sait à l'avance quelle bande il va utiliser bientôt, il peut lui aussi s'il le désire imposer une précéance à l'unité qu'il va utiliser.

Il convient évidemment d'utiliser cette possibilité à bon escient et ne pas donner par exemple la priorité à différentes unités d'un même canal par des statements successifs, car, si le canal est occupé au moment où la première requête est faite, il

est probable que la seconde annulera la première, avant que le canal ait eu le temps de la satisfaire.

CALL CLERR (IUNIT) Enlève le signal d'erreur mis par le système, l'obligeant ainsi à continuer son travail sur une unité mise en attente (voir description du travail du système)

CALL CLFIL (IUNIT) Enlève le signal d'EOF mis par le système, l'obligeant ainsi à continuer le travail sur une unité mise en attente (voir description du travail du système)

CALL CLERFL (IUNIT) Enlève les signaux d'erreur et d'EOF (combinaison de CLERR et CLFIL)

CALL SETBIN (IUNIT) Permet de changer le mode d'une unité de lecture de BCD à BIN dans devoir fermer puis rouvrir la file.

Cette routine ne peut être utilisée que soit au moment de la lecture d'un E.O.F. par le programme principal, soit après un BACKSPACE.

CALL SETBCD (IUNIT) Permet de changer le mode d'une unité de lecture (de BIN à BCD) sans devoir fermer puis rouvrir la file.

Les restrictions sont les même que pour SETBIN.

CALL ACT (IUNIT) Permet de remettre en activité une unité mise momentanément en attente après un backspace.

CALL BKFILE (IUNIT) Effectue la fermeture de l'unité - CLOSE - puis un "backspace file".

CALL UNLOAD (IUNIT) Effectue la fermeture de l'unité - CLOSE - puis une remise au point de chargement et mise hors service de l'unité.

Les routines suivantes ne peuvent être appelées que en FAP.

READ

TSX READ, 4

PZE UNIT

PZE REC, ,N

où UNIT est l'adresse d'un mot contenant en décréement le numéro

logique de l'unité à "lire" (nombre en virgule fixe type FORTRAN)

REC est l'adresse de la zone de mémoire où le record désiré doit être transféré

N est le nombre de mots à transférer (EXIT si ce nombre est supérieur à la longueur de l'enregistrement)

READ ne rend le contrôle au programme appelant que lorsque l'enregistrement désiré a été transféré réellement dans REC; il libère la partie du buffer qui contenait cet enregistrement pour la remettre à disposition du canal.

Note : READ transmet toujours au moins 1 mot

WRITE

TSX WRITE,4
PZE UNIT
PZE REC,,N

où UNIT, REC et N ont la même signification que ci-dessus (READ) sauf que cette fois les N mots contenus dans REC sont sauvés dans le buffer en attendant leur écriture sur bande.

RELEAR

TSX RELEAR,4
PZE UNIT
PZE XXXXX,,0

où UNIT voir READ

XXXXX ce que l'on veut (pas pris en considération)

0 zéro

RELEAR est équivalent à READ sans transmission. En conjonction avec GET il permet d'épargner la zone REC en travaillant directement dans la zone de BUFFER

RELEAW

TSX RELEAW,4
PZE UNIT
PZE XXXXX,,N

Sur une unité en écriture permet d'écrire une partie de buffer dont l'adresse a été obtenue au paravant par un CALL GET.

XXXXX n'est pas pris en considération

N est le nombre réel de mots à écrire (il doit être plus petit ou égal à la longueur de l'enregistrement maximum spécifié dans OPEN.

RELEAW est équivalent à WRITE lorsque le programmeur s'est chargé lui-même du remplissage de la zone de buffer; il permet d'épargner une zone REC.

TEST

TSX TEST,4

PZE UNIT (voir READ)

TRA (RETOUR D'ERREUR)

TRA (RETOUR SUR END OF FILE en lecture)

XXX retour normal

Après un READ, WRITE ou GET, TEST analyse le FLAG de l'unité et effectue un retour

2,4 en cas d'erreur permanente (en lecture seulement)

3,4 en cas d'end of file si l'unité est en lecture (en écriture le système se charge de faire changer de bande à l'opérateur en cas de end of tape)

4,4 si tout est normal.

(EFT) (BSR) (RWT) ont le même mode d'appel que la routine de même nom en FORTRAN, c'est à dire

CAL UNIT (numéro de l'unité en décrémentation dans l'accumulateur)

TSX (XXX),4

(EFT) effectue le travail éventuel restant à exécuter sur l'unité, puis y écrit un End of File.

OPEN devra être utilisé avant de pouvoir lire ou écrire sur l'unité.

(RWT) effectue le travail éventuel restant à exécuter sur l'unité puis la remet au load point. OPEN devra être utilisé avant de pouvoir lire ou écrire sur cette unité.

(BSR) positionne l'unité comme le programmeur le désire, mais en plus signale au système de ne plus activer cette unité jusqu'à nouvel ordre.

L'activation nouvelle pourra être faite soit par un appel à ACT, soit par un READ, un WRITE ou un GET. En lecture, l'usage de ACT permet de gagner du temps en réactivant au plus tôt l'unité de façon à ce que un enregistrement soit prêt au moment d'un READ ultérieur.

FONCTIONNEMENT DU SYSTEME

Le système FIOT utilise des zones mises à la disposition par le programmeur (BUFFER) pour y lire à l'avance les enregistrements qui seront utilisés par le programme principal ou y sauver rapidement l'information qui devra être écrite sur bande.

Tous les renseignements concernant l'état d'avancement des travaux du système sont contenus dans 4 tables que nous décrivons ci-après.

a) (IOU) ou (TPUN)

Cette table est constituée par la subroutine remplaçant la (IOU) normale; elle est organisée comme suit : une mémoire par unité logique appellable (de 1 à N), les unités cartes n'apparaissent pas dans cette table, La mémoire relative à une unité contient les information suivantes:

PZE XX,T,SYSF

où SYSF est l'adresse d'une mémoire contenant en adresse, l'adresse du U.C.W.* correspondant. Pour les fonctions propres à IBSYS, SYSF contient l'adresse d'une SYSUNIT; pour les unités n'ayant pas une fonction assignée, SYSF contient l'adresse d'une autre mémoire de la table qui, elle, pointe sur l'U.C.W.

* Voir IBSYS - IOEX

La partie adresse sera utilisée pour mémoriser l'adresse de la zone de lecture ou d'écriture à utiliser au prochain mouvement de l'unité. La partie tag contiendra le mode d'utilisation 0 = BIN, 7 = BCD.

Le préfixe contiendra l'usage (écriture ou lecture)

+ = lecture, - = écriture

Le préfixe et le tag sont placés lors de l'ouverture (OPEN) et le décrétement est initialisé au même moment SETBIN et SETBCD permettent de modifier le préfixe. Le décrétement est mis à jour par la "select moins" à la fin de chaque mouvement sur l'unité.

La mémoire d'entrée (IOU) contient le nombre d'unités définies. La table peut aisément être recompilée pour l'adapter à une convention différente au sujet de la relation entre numéro logique et fonction du système.

On remarquera que la mémoire relative à un numéro logique donné peut s'obtenir par index direct à partir de l'entrée de la table.

b) TBFLG

Cette table constitue une entrée du système; elle contient une mémoire par unité logique assignable.

Chacune de ces mémoires contient:

- en adresse, l'adresse de la mémoire FLAG correspondante (voir OPEN)
- en décrétement, la longueur maximum des enregistrements acceptables (LREC de OPEN)
- le préfixe contient le statut de l'unité; il mémorise si une erreur permanente (1) ou un end of file (2) a été rencontré sur l'unité; une unité portant un 1 ou un 2 en préfixe n'est plus activable tant qu'un CLERR ou CLFIL ou CLERFL n'a pas été commandé par le programme principal. Si par un statement READ INPUT TAPE ou READ TAPE on tente de lire un E.O.F. ou un enregistrement erroné le programme se termine par EXEM. Si on désire passer outre telles conditions il convient d'obtenir le flag par un GET, de le tester, de

faire ensuite un CALL RELEAR (IUNIT,0) puis un CLFIL ou CLERR ou CLERFL pour permettre à IOT de continuer la lecture sur l'unité.

- Le tag est normalement 0; il peut être mis à 7 dans deux circonstances:

1° Si le système s'est aperçu qu'il n'a plus momentanément à s'occuper de cette unité (les mémoires de la zone de buffer de l'unité sont entièrement à la disposition du programme principal en d'autres termes encore: elles sont remplies d'enregistrements non encore utilisés en lecture ou ne contiennent plus aucun enregistrement à écrire).

2° Un backspace a été effectué sur cette unité

Un tag = 7 met l'unité "dormante"; pour la réactiver il faut utiliser ACT ou READ ou WRITE ou GET ou RELEAR

c) TBPRC

Cette table appartient au système et ne peut être touchée de l'extérieur; elle comporte une mémoire par unité logique. En adresse se trouve le nombre de zone de lecture ou d'écriture à disposition du canal (c'est à dire libre pour y lire de nouveaux enregistrements ou contenant des enregistrements en attente d'être écrits).

Le décrétement contient le nombre total de zone de lecture ou d'écriture que OPEN a pu construire dans le buffer mis à sa disposition.

Le tag contient le numéro de canal auquel appartient l'unité.

La routine interne de prééance à laquelle se réfère la

"select moins" tend à rendre la partie adresse inférieure à la moitié de la partie décrétement et si possible naturellement à annuler toutes les parties adresses.

d) TBADD

Cette table contient une mémoire par unité logique adressable; la partie adresse pointe sur la "commande" associée à

la prochaine zone qu'il faudra mettre à disposition du programme principal lorsqu'il demandera à lire où à écrire.

La partie décrétement indique l'adresse de la zone où le dernier READ ou le dernier WRITE a été effectué. (Note: cette partie pourrait être facilement récupérée si une amplification du système s'avérait nécessaire, elle a été conservée pour faciliter éventuellement des recherches en cas de mauvais fonctionnement d'un programme).

Le préfixe est utilisé pour indiquer si un "GET" a déjà été fait sur l'enregistrement actuellement en cours (Préfixe = 2).

Une cinquième table (une mémoire par canal utilisé - 2 -)

indique l'état du canal et l'unité éventuelle qui a priorité.

Si l'adresse est 0 le canal est actuellement au repos; si l'adresse est \neq 0 le canal est occupé.

Si le décrétement est nul, au moment de la "select moins" la routine de détermination des précéances doit être consultée pour fixer le numéro de l'unité à activer, si le décrétement est non nul, il contient le numéro logique de l'unité qu'il convient (sous contrainte extérieure) d'essayer d'activer en premier lieu sur ce canal.

L'activation sera cependant refusée si le préfixe et le tag de la mémoire relative à cette unité dans TBFLAG ne sont pas tous deux nuls.

SETPR permet de mettre le numéro d'une unité dans le décrétement de la mémoire de précéance du canal auquel elle appartient.

Lors de l'appel de OPEN le système répartit les mémoires de BUFFER en autant de zones de lecture ou d'écriture de longueur LREC qu'il est possible en associant à chacune de ces zones une mémoire de "commande".

La structure des memoires de commandes est la suivante:

Signe + la zone est à disposition du canal

- la zone à disposition du programme.

Ceci signifie + en lecture la zone peut être utilisée pour

y lire un enregistrement de la bande

en écriture la zone contient des informations

qui n'ont pas encore été écrites.

- en lecture la zone contient un enregistrement non encore utilisé par le programme principal
- en écriture la zone est libre pour y stocker de l'information qui devra être écrite.

Positions 1 et 2 = 3 lorsque le signe est + (IORT)

= 0, 1 ou 2 lorsque le signe est -

0 signifie que tout est normal

1 l'enregistrement lu a donné erreur permanente

2 l'enregistrement correspondant était un E.O.F.

en adresse: l'adresse de la zone de lecture ou d'écriture associée;

en décrétement: si l'unité est en lecture: si le signe - , la longueur de la zone (LREC)
: si le signe est + la longueur effective du record lu.

En écriture si le signe est - la longueur de la zone

si le signe est + la longueur effective de l'information à écrire

OPEN initialise également les mémoires concernant l'unité citée dans les diverses tables.

Si l'unité est en lecture et si le canal est inactif, elle active le canal.

Les routines READ, WRITE, GET, RELEAW, RELEAR en plus de leurs fonctions déjà décrites, réactivent éventuellement le canal correspondant à l'unité.

Le select plus (appelé par ACTIV dans IOEX) commande effectivement l'unité qui lui a été désignée.

Le select moins vérifie le résultat de l'opération sur bande, tient à jours les tables d'adresses et les indications d'erreurs ou d'end of file puis si une préséance existe sur le canal active l'unité désignée si elle est activable si non, s'adresse à la routine de préséance pour en obtenir le numéro de la prochaine unité à activer sur le canal. S'il n'y a plus rien à faire elle met à zéro la mémoire correspondante à ce canal en TBCHN.

Note: l'index 2 transporte le numéro d'unité

l'index 1 indique le canal

La plupart des entrées utilisent INIT pour sauver les index et préparer l'index 2.

SELCH prépare l'index 1

La mémoire du transfert vecteur correspondant à (TPUN) est modifiée lors de la première entrée à OPEN. Un tag = 2 y est placé.

Une équivalence est faite entre le symbole TAPUNT et cette mémoire. Toutes les références à IOU sont faites par adressage indirect de TAPUNT avec le numéro logique de l'unité dans l'index 2.

Liste des routines fonctionnant avec IOT

LABEL	ENTREES	LONGUEUR EN OCTAL	ROUTINES APPELLEES
IOTV3	OPEN READ RELEAR WRITE RELEAW GET CLOSE CLOSAL (BSR) (RWT) (EFT) UNLOAD BKFILE SETPR TEST CLERR CLFIL CLERFL ACT SETBCD SETBIN TBADD TBPRC TBFLG	1145	(IOU), (EXE), (TPUN)
IOBMOD	(IOB) (SET) (BUF) (EXB)	101	SETPR, GET, (TES), TBFLG
IOSMOD	(IOS)	1	
IOUMOD	(IOU) (TPUN)	24	
RERMOD	(RER)	30	TEST, (IND), GET, (EXE)

LABEL	ENTREES	LONGUEUR EN OCTAL	ROUTINES APPELEES
STBMOD	(STB) (WLR)	37	(IOB), RELEAW, (EXB)
STHMOD	(STH), (STHM) (STHD)	52	(IOH), WRITE
TSBMOD	(TSB), (RLR), (IND)	76	(TES), (IOB), SETPR, (EXE), (SET), GET, (RER) RELEAR, (EXB)
TSHMOD	(TSH), (TSHM)	43	(IOH), TBFLG), READ, TEST, (EXE)
EXEMOD	(EXEM)	657	(TPUN), EXSEL, EXIT, EXITFN, (IOS)
EXTMOD	EXIT, EXSEL EXITFN, FLGXIT	210	CLOSAL, (EFT), (DUM)
BSTMOD	(BST)	35	(TPUN), TBFLG, (BSR)
CHNMOD	CHAIN	342	CLOSAL, EXSEL, EXIT, (DUM)
DUPMOD	DUMP, PDUMP	1	
SPHMOD	(SPH)	16	(IOH)
BCDRD	BCDRD, (CSH)	113	(IOH), GET, TEST, RELEAR, (EXE)
BCDWR	BCDWR, FINWR	116	(IOH), GET, TBFLG, RELEAW
CHNMB5	CHAIN	317	CLOSAL, EXSEL, EXIT
DUMOD	(DUM)	1	(DUM)
PAUSE	PAUSE	11	
PLTMOD	PLOT, PLTIR, FINIM, FINTRA	1031	(EFT), (STB), (WLR), (EXE)
SKFILE	SKFILE	76	(TPUN), TBADD, TBPRC, TBFLG, (IOU), CLERFL, EXSEL, (EXE)
STHBLC	(STH), (STHM), (FHW), SETOUT	234	GET, TBFLG, RELEAW, (DUM), (IOH), WRITE, (FIL), EXIT
STHMCL	(STH), (STHM), (STHD)	100	(IOH), WRITE, (FIL), EXIT

ROUTINES A INSERER

Nous donnons ci-dessous la liste des decks binaires requis par les divers statements d'entrée-sortie de FORTRAN en plus des decks IOTV3, EXEMOD, DUMOD, IOUMOD et IOSMOD qui sont toujours nécessaires.

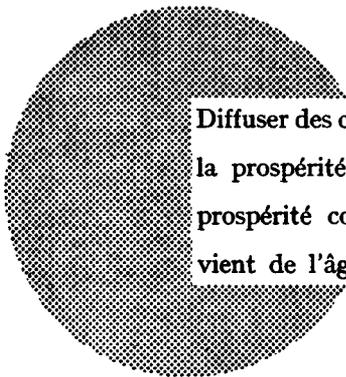
READ INPUT TAPE	TSHMOD (et 9IOH normal de FORTRAN)
WRITE OUTPUT TAPE	STHMOD ou STHBLC (et 9IOH normal de FORTRAN)
READ TAPE	TSBMOD IOBMOD RERMOD
WRITE TAPE	STBMOD IOBMOD
PRINT	SPHMOD (et 9IOH normal de FORTRAN plus 9SCH normal de Ispra)
BACKSPACE	BSTMOD
ENDFILE	aucune routine spéciale
REWIND	
READ	(n'existe pas en fait et ne peut être uti- lisé que conjointement à BCDRD (CSH) est inclu des BCDRD)

Liste des decks binaires

LABEL	Remarques
IOTV3 : ROUTINE IOT	Toujours nécessaire
IOBMOD: (IOB)	READ ou WRITE TAPE
IOSMOD: (IOS)	READ ou WRITE OUTPUT TAPE ou PUNCH
IOUMOD: (IOU)	Toujours utilisée ou PRINT
RERMOD: (RER)	READ TAPE
STBMOD: (STB)	WRITE TAPE
STHMOD: (STH)	WRITE OUTPUT TAPE
TSBMOD: (TSB)	READ TAPE
TSHMOD: (TSH)	READ INPUT TAPE
EXEMOD: (EXE), (EXEM)	Toujours nécessaire
BSTMOD: (BST)	BACKSPACE
EXTMOD: EXIT	Toujours nécessaire
CHNMOD: (CHAIN)	
DUMMOD: (DUM)	Toujours utilisée
DUPMOD: DUMP	
EUSCH: (SCH)	Statement PUNCH - ECRITURE SUR B4
SPHMOD: (SPH)	Statement PRINT
BCDRD: BCDRD	
BCDWR: BCDWR	
CHNB5M: CHAIN	Sur 25 seulement
PAUSE: PAUSE	
PLTMOD: PLOT	Pour CALCOMP et 1401
SXFILE: SKFILE	
STHBLC: (STH)	Spéciale faisant le blocking de la bande de sortie normale et compte des lignes
STHMBL: (STH)	Spéciale faisant le blocking sans compte de ligne
STHMCL: (STH)	Normale avec compte des lignes

Note Les routines avec compte de ligne doivent toujours être utilisées à Ispra.

CHN5B : CHAIN Cette routine n'utilise pas le système IOT et peut être utilisée pour fabriquer un chargeur appelant une bande quelconque chargée sur B5.



Diffuser des connaissances c'est distribuer de la prospérité — j'entends la prospérité collective et non la richesse individuelle — et cette prospérité contribue largement à la disparition du mal qui nous vient de l'âge des ténèbres.

Alfred Nobel

BUREAUX DE VENTE

Tous les rapports Euratom sont vendus dans les bureaux suivants, aux prix indiqués au verso de la couverture (lors de la commande, bien indiquer le numéro EUR et le titre du rapport, qui figurent sur la couverture).

PRESSES ACADEMIQUES EUROPEENNES

98, Chaussée de Charleroi, Bruxelles 6

Banque de la Société Générale - Bruxelles
compte N° 964.558,

Banque Belgo Congolaise - Bruxelles
compte N° 2444.141,

Compte chèque postal - Bruxelles - N° 167.37,

Belgian American Bank and Trust Company - New York
compte No. 22.186,

Lloyds Bank (Europe) Ltd. - 10 Moorgate, London E.C.2,
Postcheckkonto - Köln - Nr. 160.861.

OFFICE CENTRAL DE VENTE DES PUBLICATIONS DES COMMUNAUTES EUROPEENNES

2, place de Metz, Luxembourg (Compte chèque postal N° 191-90)

BELGIQUE — BELGIË

MONITEUR BELGE
40-42, rue de Louvain - Bruxelles
BELGISCH STAATSBLAD
Leuvenseweg 40-42 - Brussel

GRAND-DUCHE DE LUXEMBOURG

OFFICE CENTRAL DE VENTE
DES PUBLICATIONS DES
COMMUNAUTES EUROPEENNES
9, rue Goethe - Luxembourg

DEUTSCHLAND

BUNDESANZEIGER
Postfach - Köln 1

ITALIA

LIBRERIA DELLO STATO
Piazza G. Verdi, 10 - Roma

FRANCE

SERVICE DE VENTE EN FRANCE
DES PUBLICATIONS DES
COMMUNAUTES EUROPEENNES
28, rue Desaix - Paris 15^e

NEDERLAND

STAATSDRUKKERIJ
Christoffel Plantijnstraat - Den Haag

EURATOM — C.I.D.
51-53, rue Belliard
Bruxelles (Belgique)